REMARKS

Claims 1-26 are pending. Claims 1, 11, 13, 19, 23, 25, and 26 are amended. Support for the amendments may be found on at least page 11, line 5, to page 12, line 4, of the originally filed specification. Reconsideration of the claims is respectfully requested in view of the following remarks.

I. Priority

The Office Action notes that the Claim for Foreign Priority filed 3/11/2004 inadvertently claims priority to the instant application. The Examiner is correct in interpreting the Claim for Foreign Priority as being directed to Canadian Application No. 2,453,605. Applicants hereby claim priority to Canadian Application No. 2,453,605. If the Examiner believes a corrected Claim for Priority or a replacement declaration is necessary, then Applicants will file such. Applicants kindly request the Examiner to direct Applicants in the next Office Action as to how to correct the Claim for Foreign Priority.

II. 35 U.S.C. § 112, Second Paragraph, Allegedly Indefinite Subject Matter

The Office rejects claims 11 and 23 under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicants regard as the invention. Applicants amend claims 11 and 23 to more clearly recite the subject matter that Applicants regard as the invention. Therefore, Applicants respectfully request withdrawal of the rejection of claims 11 and 23 under 35 U.S.C. § 112, second paragraph.

III. 35 U.S.C. § 103(a), Alleged Obviousness of Claims 1-9, 12-21, and 24-26

The Office rejects claims 1-9, 12-21, and 24-26 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Upson et al.* (U.S. Patent No. 5,694,578) in view of *Banning et al.* (U.S. Patent No. 5,421,008). Applicants respectfully traverse this rejection.

Upson teaches a computer-implemented method and apparatus for converting data according to a selected data transformation. *Upson* states:

> The conversion system allows a user to convert data in a given structure into a desired data structure. Initially, the user constructs a visual graphical input template that describes the structure of the input data to the present invention. Only those pieces of data that are necessary for construction of the output structure need be described. Thus, the present invention can be operated in such a manner that uninteresting pieces of an input data structure are ignored.
>
> Thereafter, the user constructs a visual graphical output template which describes the desired structure of the output data. Finally, the user describes the manner in which the output data structure is to be constructed from the input data structure. Such a description is accomplished by selecting pieces of the input data template and "assigning" those pieces to the corresponding pieces of the output template. The present invention will then evaluate the description and generate a data chopping or scribe module 94 which will carry out the data conversion specified by the user. The end result is a data transform.

Upson, col. 4, lines 35-53. Thus, *Upson* teaches a data transform that is generated graphically (visual based module). *Upson* also states:

> The data transform librarian 88 allows for data transforms to be stored within it. In addition to receiving the p-code fragments 76 and reference table 78 from a given data transform, the data transform librarian 88 is also connected to the visual manager 72 for load/save instances. Each complete data transform contains three pieces: the textual description 86, the p-code fragments 76, and the reference tables 78 of the data transform. A complete data transform will require all three pieces in order for the data scribe module 94 to execute. An incomplete data transform may, however, be stored within the data transform librarian 88. An incomplete data transform may result from work in progress or libraries of clip-templates, which are useful templates or template components that can be utilized.

Upson, col. 6, lines 7-20. Thus, *Upson* teaches that all data transforms are stored in the same form. In other words, all data transforms are visual based and stored in the library in a common form.

In contradistinction, the present invention, as recited in claim 1, receives an instruction for selecting the set of transformation modules from a memory and converts each of the set of transformation modules to a common model format. The set of modules has a first transformation module being a language constructed module and a second transformation module being a visually constructed module. *Upson* does not teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module. Rather, all of the data transforms in *Upson* are visually constructed. See *Upson*, col. 4, lines 35-53. Furthermore, there is no need to convert the data transforms of *Upson* into a common model format, because the data transforms of *Upson* are already stored in a common form.

Furthermore, the present invention, as recited in claim 1, recites that one of the first module or the second module references the other of the first module or the second module. *Upson* does not teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module. See *Upson*, col. 4, lines 35-53. Therefore, *Upson* cannot teach a language constructed transform module referencing a visually constructed transform module or *vice versa*.

The Office Action cites *Banning* as allegedly teaching that text based instructions could be saved in a common data model along with visual based instructions at col. 8, lines 31-35, which states:

> One aspect of the invention was the recognition that bidirectional translation or conversion between text based SQL query statements and graphically based visual query representations requires the creation and use of a common data structure.

Thus, *Banning* teaches that translation of text based SQL query statements, in their executable form, and graphically based visual query representations, in their editable form in a graphical user interface, requires a common data structure. Applicants note that the above cited portion of *Banning* does not teach or fairly suggest converting data transform modules of different types to a common model. Rather, *Banning* teaches throughout the disclosure that all SQL statements are created visually. See *Banning*, Abstract, which states:

> A method, system and program providing graphical queries and direct manipulation of a database is disclosed.

> In a preferred form, tables and lists are configured from a database to define a common data structure. Additional, dynamic data structures are employed based on information entered by a user to define various relationships between the dynamic data and the database information. The system employs a graphical query interface and a relational database to provide an ergonomic, natural interface for a database user.

Banning does appear to recognize that translating between the graphical form and its corresponding SQL text requires a common model. See *Banning*, col. 8, lines 31-35. However, *Banning*, like *Upson*, fails to teach or suggest receiving an instruction for selecting the set of transformation modules from a memory and converting each of the set of transformation modules to a common model format.

Furthermore, *Banning*, like *Upson*, fails to teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module, because *Banning* teaches only visually constructed SQL statements. See *Banning*, Abstract.

Upson and *Banning*, taken individually or in combination, fail to teach or fairly suggest each and every claim feature. Therefore, the proposed combination of *Upson* and *Banning* does not render claim 1, for example, obvious. Independent claims 13, 25, and 26 recite subject matter addressed above with respect to claim 1 and are allowable for similar reasons. Because claims 2-12 and 14-24 depend from claims 1 and 13, the same distinctions between *Upson* and *Banning* and claims 1 and 13 apply for these claims. In addition, claims 2-12 and 14-24 recite additional combinations of features not taught or suggested by *Upson* and *Banning*.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1-9, 12-21, and 24-26.

IV. 35 U.S.C. § 103(a), Alleged Obviousness of Claims 10, 11, 22, and 23

The Office rejects claims 10, 11, 22, and 23 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Upson* in view of *Banning* and further in view of *Aho et al.* ("Compilers: Principles, Techniques, and Tools"). Applicants respectfully traverse this rejection.

With respect to claims 10 and 22, the Office Action acknowledges that *Upson* and *Banning* fail to teach or suggest that the common model format is generic for suitable generation of the executable version for a selected one of a plurality of runtime environments for the data transform engine. The Office Action alleges that *Aho* teaches an intermediate representation that is used for generation for suitable executables at page 12, which states:

> After syntax and semantic analysis, some compilers generate an explicit intermediate representation of the source program. We can think of this intermediate representation as a program for an abstract machine. This intermediate representation should have two important properties: it should be easy to produce, and easy to translate into a target program.

Aho, bottom of page 12. While *Aho* does appear to teach an intermediate representation of a source program in the cited portion above, *Aho* does not teach or fairly suggest a common **model** for a **data transform module**.

Furthermore, with respect to claims 10, 11, 22, and 23, *Aho*, like *Upson* and *Banning*, fails to teach or fairly suggest receiving an instruction for selecting the set of transformation modules from a memory and converting each of the set of transformation modules to a common model format, wherein the set of transform modules includes a language constructed module and a visually constructed module. *Upson*, *Banning*, and *Aho* taken individually or in combination, fail to teach or fairly suggest each and every claim feature. Therefore, the proposed combination of *Upson*, *Banning*, and *Aho* does not render claims 10, 11, 22, and 23 obvious.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 10, 11, 22, and 23.

V. Conclusion

It is respectfully urged that the subject application is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: December 28, 2007

Stephen R. Tkacs
Reg. No. 46,430
WALDER INTELLECTUAL PROPERTY LAW, P.C.
P.O. Box 832745
Richardson, TX 75083
(214) 722-6422
AGENT FOR APPLICANTS